МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)

**Закамский региональный конкурс**

**«IT-разработка 2020» для школьников и студентов**

**среднего профессионального образования**

Секция «IT-разработка Web-конструирование»

**Тема: Разработка web-сайта, посвященного логическим задачам и играм** **с использование современных способов   
web-разработки.**

Выполнил: Ученик 9 «Д» класса,

МБОУ «Гимназия №26»

Данилов Тимофей Николаевич

Научный руководитель:

учитель информатики

МБОУ «Гимназия №26»

Волошина Гульшат Мунировна

Набережные Челны – 2020

Описание разработки

Цель работы: разработать сайт, посвященный логическим задачам и играм с использование современных способов web-разработки.

Выполнение данной работы состояло из нескольких этапов:

1. Выбор задач
2. Создание дизайна
3. Адаптивная вёрстка
4. Добавление скриптов
5. Публикация в интернете

Перед началом работы необходимо подобрать задачи и игры, которые нужно будет реализовать на сайте. Это очень важно, ведь чем интереснее будут задачи, тем лучше будет реакция пользователей.

При выборе задач, я упирался на то, чтобы они были рассчитаны не на хорошие знания в математике, а на нестандартное мышление.

Немаловажно в создании сайта – это качественная разработка дизайна и оформления, ведь чем удобнее и понятнее будет дизайн, тем проще пользователям будет с ним работать. Для создания дизайна я использовал программу Adobe XD.

Следующий этап – создание структуры гипертекстового документ, то есть вёрстка. Мы уже создали дизайн, но процесс усложняется тем, что в 21 веке большинство людей заходят на сайты не с компьютера, а с телефона, поэтому вёрстку нужно было сделать адаптивной, чтобы на любом устройстве выглядел сайт приятно.

Так как сайт посвящен логическим задачам и играм, обойтись без дополнительной логики не получится, для этого пришлось написать множество скриптов на языке TypeScript. Пример кода реализации игры Математико:

import { Component, OnInit } from '@angular/core';

import { animate, style, transition, trigger } from '@angular/animations';

@**Component**({

  selector: 'bb-mathematico',

  templateUrl: './mathematico.component.html',

  styleUrls: ['./mathematico.component.scss'],

  animations: [

**trigger**('enterAnimation', [

**transition**(':enter', [

**style**({ transform: 'scale(0)', opacity: 0 }),

**animate**('150ms', **style**({ transform: 'scale(1)', opacity: 1 }))

      ]),

**transition**(':leave', [

**style**({ transform: 'scale(1)', opacity: 1 }),

**animate**('150ms', **style**({ transform: 'scale(0)', opacity: 0 }))

      ])

    ])

  ]

})

export class **MathematicoComponent** implements **OnInit** {

  current = 0;

  table: number[][];

  buttons: null[][];

  numbers: number[];

  size = 5;

  left: number;

  ended: boolean;

  score: number;

  combinations: {

    name: string;

**func**: (ns: number[]) => boolean;

    row: number;

    diagonal: number;

  }[] = [

    {

      name: 'За 2 одинаковых числа',

**func**: (ns: number[]) => {

        for (let i = 0; i < ns.length - 1; i++) {

          if (this.**allEqual**(ns.**slice**(i, i + 2))) return true;

        }

        return false;

      },

      row: 10,

      diagonal: 20

    },

    {

      name: 'За 2 пары одинаковых чисел',

**func**: (ns: number[]) => {

        let one: boolean = false;

        for (let i = 0; i < ns.length - 1; i++) {

          if (this.**allEqual**(ns.**slice**(i, i + 2))) {

            if (one) return true;

            one = true;

            i++;

          }

        }

        return false;

      },

      row: 20,

      diagonal: 30

    },

    {

      name: 'За 3 одинаковых числа',

**func**: (ns: number[]) => {

        for (let i = 0; i < ns.length - 2; i++) {

          if (this.**allEqual**(ns.**slice**(i, i + 3))) return true;

        }

        return false;

      },

      row: 40,

      diagonal: 50

    },

    {

      name: 'За 3 одинаковых числа и два других одинаковых числа',

**func**: (ns: number[]) => {

        return (

          (ns[0] === ns[1] && ns[1] === ns[2] && ns[3] === ns[4]) ||

          (ns[0] === ns[1] && ns[2] === ns[3] && ns[3] === ns[4])

        );

      },

      row: 80,

      diagonal: 90

    },

    {

      name: 'За 4 одинаковых числа',

**func**: (ns: number[]) => {

        for (let i = 0; i < ns.length - 3; i++) {

          if (this.**allEqual**(ns.**slice**(i, i + 4))) return true;

        }

        return false;

      },

      row: 160,

      diagonal: 170

    },

    {

      name:

        'За 5 последовательных чисел, но не обязательно по порядку расположенных',

**func**: (ns: number[]) => {

        for (let i = 1; i < this.size; ++i)

          if (ns[0] + i !== ns[i]) return false;

        return true;

      },

      row: 50,

      diagonal: 60

    },

    {

      name: 'За три раза по 1 и два раза по 13',

**func**: (ns: number[]) => {

        return (

          this.**allEqual**(ns.**slice**(0, 3), 1) && this.**allEqual**(ns.**slice**(0, 3), 13)

        );

      },

      row: 100,

      diagonal: 110

    },

    {

      name:

        'За числа 1, 13, 12, 11 и 10, но не обязательно по порядку расположенных',

**func**: (ns: number[]) => {

        return (

          ns[0] === 1 &&

          ns[1] === 10 &&

          ns[2] === 11 &&

          ns[3] === 12 &&

          ns[4] === 13

        );

      },

      row: 150,

      diagonal: 160

    },

    {

      name: 'За 4 единицы',

**func**: (ns: number[]) => {

        return this.**allEqual**(ns.**slice**(0, 4), 1);

      },

      row: 200,

      diagonal: 210

    }

  ].**sort**((a, b) => a.row - b.row);

  private **allEqual**(arr: any[], n: number = null) {

    if (n !== null) return arr.**every**((val, i, arr) => val === n);

    return arr.**every**((val, i, arr) => val === arr[0]);

  }

  constructor() {}

**ngOnInit**(): void {

    this.**load**();

  }

**load**() {

    this.left = this.size \*\* 2;

    this.ended = false;

    this.**fillNumbers**();

    this.**fillButtons**();

    this.**fillTable**();

    this.**getCurrent**();

  }

**fillButtons**() {

    this.buttons = [];

    for (let i = 0; i < this.size; ++i) {

      this.buttons.**push**(new **Array**(this.size).**fill**(null));

    }

  }

**getCell**(x: number, y: number): number | '' {

    return this.table[x][y] ? this.table[x][y] : '';

  }

**fillTable**() {

    this.table = [];

    for (let i = 0; i < this.size; ++i) {

      this.table.**push**(new **Array**(this.size).**fill**(0));

    }

  }

**fillNumbers**() {

    this.numbers = [];

    for (let i = 1; i <= 13; ++i)

      this.numbers = this.numbers.**concat**(new **Array**(4).**fill**(i));

  }

**set**(x: number, y: number) {

    if (this.table[x][y]) return;

    this.table[x][y] = this.current;

    this.**getCurrent**();

    this.left--;

    if (this.left <= 0) {

      this.**end**();

    }

  }

**end**() {

    this.ended = true;

    const rows = [];

    const diagonals = [];

    this.score = 0;

    for (let i = 0; i < this.table.length; ++i) {

      let row = [];

      for (let j = 0; j < this.table[i].length; ++j) {

        row.**push**(this.table[j][i]);

      }

      rows.**push**(this.table[i].**slice**());

      rows.**push**(row);

    }

    let diag = [];

    for (let i = 0; i < this.table.length; ++i) {

      diag.**push**(this.table[i][i]);

    }

    diagonals.**push**(diag);

    diag = [];

    for (let i = 0; i < this.table.length; ++i) {

      diag.**push**(this.table[i][this.size - i - 1]);

    }

    diagonals.**push**(diag);

    rows.**forEach**(v => {

      this.score += this.**calculate**(v);

    });

    diagonals.**forEach**(v => {

      this.score += this.**calculate**(v, true);

    });

  }

**getCurrent**() {

    if (this.numbers.length <= 0) return;

    this.current = this.numbers.**splice**(

      Math.**floor**(Math.**random**() \* this.numbers.length),

      1

    )[0];

  }

**calculate**(ns: number[], diag: boolean = false): number {

    let score = 0;

    ns.**sort**((a, b) => a - b);

    for (let i = this.combinations.length - 1; i >= 0; --i) {

      const v = this.combinations[i];

      if (v.**func**(ns.**slice**())) {

        score += diag ? v.diagonal : v.row;

        break;

      }

    }

    return score;

  }

}

Все этапы программирования создавались в программе Visual Studio Code. Эта программа является бесплатным редактором кода от Microsoft. Весь сайт построен на фреймворке Angular с использование библиотек TypeScript и Scss.

Последним этапом является публикация сайта в интернет, но перед этим нужно собрать проект, ведь языки TypeScript и Scss браузер понять не может. Для этого я использовал библиотеку Webpack, которая после хорошей настройки превращает весь код на понятный браузеру, то есть написанный на языках HTML, Css и JavaSript. Помимо этого, Webpack создает версии кода на устаревших языках программирования, чтобы даже в старых браузерах сайт выглядел отлично. После этого я выбрал и настроил хостинг, подобрал домен для сайта и выложил проект в сеть. Теперь он доступен по адресу <http://break-brain.epizy.com/>

Несмотря на то что разработка сайта с использованием таких высокоуровневых средств разработки является очень трудоёмкой и требует многолетнего опыта работы, процесс был очень интересным и увлекательным.